

Training Multi-Modal LLMs through Dialogue Planning for HRI

Claudiu Daniel Hromei, Federico Borazio, Andrea Sensi,
Elisa Passone, Danilo Croce and Roberto Basili

Department of Enterprise Engineering
University of Rome Tor Vergata
Via del Politecnico 1, 00133, Rome, Italy

Abstract

Grounded natural language understanding in Human-Robot Interaction (HRI) requires integrating linguistic, visual, and world knowledge to ensure effective task execution. We propose an approach that enhances Multi-Modal Large Language Models (MLLMs) with a novel explicit dialogue planning phase, allowing robotic agents to systematically refine their understanding of ambiguous commands through structured clarification steps. This reduces hallucinations and improves task feasibility. To evaluate this approach, we introduce a novel dataset of over 1,100 annotated dialogues in English and Italian, designed for fine-tuning and assessing Multi-Modal models in HRI scenarios. Experimental results show that dialogue planning improves response accuracy and quality, and contributes to cross-lingual generalisation, enabling models trained in one language to transfer effectively to another. To the best of our knowledge, this is the first application of structured, goal-driven, and explicit dialogue planning in Multi-Modal LLMs for grounded interaction.

1 Introduction

Grounded natural language understanding in Human-Robot Interaction (HRI) presents significant challenges due to the need to integrate linguistic, visual, and world knowledge to ensure effective task execution (Kiseleva et al., 2022). In real-world scenarios, a robotic agent receives Multi-Modal inputs: a natural language command from a human user and a perceptual representation of the environment, typically in the form of visual input, such as images or derived from more complex sensors, as in (Huang et al., 2023). However, human instructions may be underspecified or ambiguous (Borazio et al., 2024), requiring an interactive dialogue where the agent actively requests missing information before taking action.

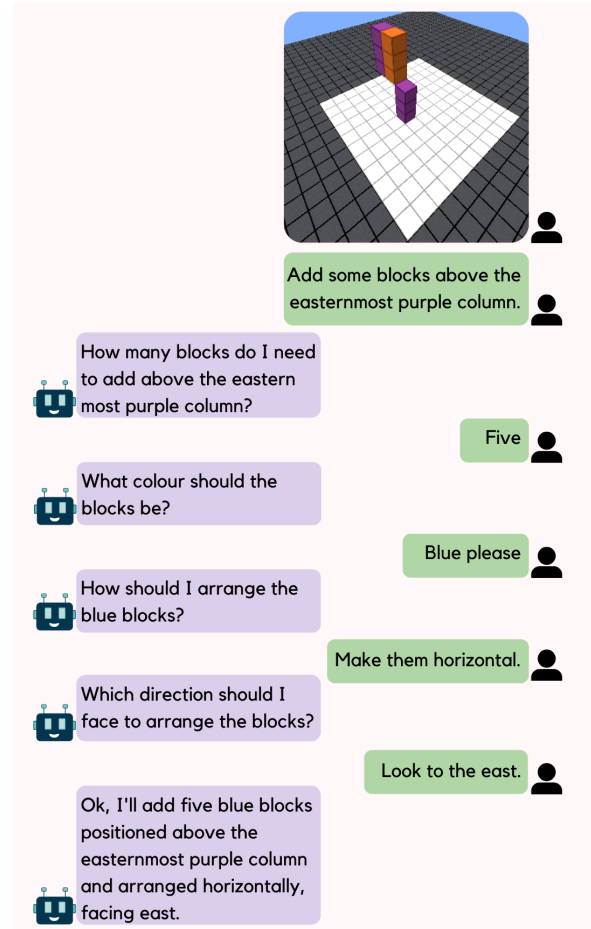


Figure 1: Example of a Multi-Modal interaction where the agent iteratively refines its understanding of a command through clarification steps before execution.

Recent advances in Multi-Modal Large Language Models (MLLMs) have enabled the joint processing of textual and visual information (Yin et al., 2024). While these models were not originally designed for HRI, they provide a promising foundation for robotic applications, allowing agents to interpret visual contexts and language simultaneously. Hromei et al., 2024 have explored the application of MLLMs in Multi-Modal environments, such as the Minecraft-inspired MM-IGLU

framework, where agents must reason over commands and visual scenes to understand if a command is executable or not.

However, standard MLLMs lack reasoning capabilities for solving specific tasks: they generate responses based on local conversational history but do not construct a structured sequence of information-seeking steps, which is crucial in HRI. This limitation risks leading to inefficient interactions, redundant queries, or hallucinated responses (Bai et al., 2024) that do not align with the environment. Recent research has demonstrated that incorporating planning mechanisms into LLMs can significantly enhance their robustness and generalisation capabilities. Techniques such as Chain-of-Thought reasoning (Wei et al., 2022) have proven beneficial in guiding models through complex decision-making tasks.

In this work, we study the role of planning within a Multi-Modal setting, focusing on grounded interaction with a robotic agent operating in an environment similar to MM-IGLU (Hromei et al., 2024; Borazio et al., 2025). Figure 1 illustrates the structured dialogue planning mechanism employed in our approach. In a Multi-Modal Human-Robot Interaction (HRI) setting, a robotic agent receives an initial command from the user along with a visual representation of the environment. However, the command may be incomplete, requiring additional clarification before it can be executed correctly. Instead of attempting an immediate response, the agent engages in an interactive process to gather missing details. At each turn, the agent processes three inputs: the current world state provided as an image, the accumulated dialogue history containing prior interactions, and the user’s latest utterance. Given this context, an MLLM generates a structured plan identifying the missing information required for task completion. Rather than providing an immediate response, the plan serves as a guide for the necessary clarifications before execution.

Once the plan is generated, it is incorporated into the ongoing interaction to guide the formulation of the next clarification question. The agent iterates through this process, dynamically updating its plan as new information emerges from the dialogue. The clarification questions systematically address ambiguities related to attributes such as the number of blocks, their colour, orientation, or precise positioning within the scene. When all necessary details have been gathered, the agent produces a final recap summarising the inferred command. This

final step allows the user to confirm the interpreted instruction before execution, ensuring alignment between the intended and understood task. To this end, benchmarks (Attanasio et al., 2024) have been created for evaluating their capabilities.

To evaluate the effectiveness of our approach, we have created a dataset of more than 1,100 dialogues in English, specifically designed for fine-tuning and evaluating Multi-Modal models in HRI scenarios¹. Although many State-of-the-Art models undergo pre-training on large-scale corpora of text and images, such pre-training does not necessarily equip them with the structured reasoning and dialogue capabilities required for interactive problem-solving.

To demonstrate the broad applicability of our approach, independent of the underlying architecture, we apply it to two State-of-the-Art Multi-Modal models, LLaVA (Liu et al., 2024) and MiniCPM (Yao et al., 2024), and demonstrate that:

1. *Explicit dialogue planning contributes substantially to system performance.* To the best of our knowledge, this is the first application of dialogue planning in Multi-Modal models. Our experimental results show significant performance gains when transitioning from a zero-shot setting to a fine-tuned setting where interaction history is utilised. Furthermore, incorporating planning further enhances results by making dialogues more efficient and reducing redundant or repetitive queries.
2. *We show that cross-lingual generalisation can be achieved through the adoption of a bilingual English and Italian dataset.* We test whether planning-based approaches improve robustness when a model trained in one language is applied to another. Our results suggest that planning facilitates generalisation: a model trained in English and applied to Italian (with planning) achieves performance comparable to a model fine-tuned directly in Italian but lacking planning capabilities.

In the remainder of the paper, Section 2 discusses related work, Section 3 describes the proposed Multi-Modal architecture and dialogue planning approach, Section 4 details experimental evaluation, and Section 5 concludes our findings.

¹All the resources and code are available at <https://github.com/crux82/MM-IGLU-Dialogues>

2 Related Work

Effective Multi-Modal dialogue planning requires integrating structured decision-making into LLMs, enhancing model outputs with external knowledge, and systematically evaluating the impact of these mechanisms. Our work builds upon three key research directions.

First, explicit planning mechanisms have been explored to enhance structured reasoning in LLMs, but primarily outside the context of HRI. Prior studies have integrated planning into LLMs for tasks such as code generation (Jiang et al., 2024), visual question answering (Scaiella et al., 2024), response structuring in textual dialogues (Li and Zhang, 2024), and incremental decision-making (Gestrin et al., 2024). While these works demonstrate the benefits of intermediate planning steps, they focus on structuring outputs in non-embodied scenarios rather than in interactive, Multi-Modal dialogue settings. A closely related line of work is DialFRED (Gao et al., 2022), which extends the ALFRED benchmark (Shridhar et al., 2020) to support dialogue-enabled agents capable of asking clarification questions when task instructions are ambiguous. Unlike our approach, which introduces explicit plan generation to structure such interactions, DialFRED equips agents with a questioning module but lacks systematic integration of planning steps to guide the dialogue evolution. In contrast, planning has also been applied in HRI, but predominantly for action execution rather than dialogue structuring. For example, some studies have investigated planning-based decision-making for task execution in embodied agents (Hao et al., 2023; Wu et al., 2023; Hromei et al., 2022a,b), where models generate step-by-step action sequences to achieve a goal. However, these approaches do not address how planning can be leveraged within dialogue itself to systematically elicit missing information before taking action.

Second, Multi-Modal dialogue systems have been enhanced through external knowledge integration, improving response coherence by leveraging perceptual and linguistic information. Previous work has focused on enriching dialogue models with implicit visual cues (Zhang et al., 2025; Hromei et al., 2023a) or persona-driven dialogue structuring (Alavi et al., 2024). These contributions highlight the benefits of injecting Multi-Modal information but do not employ explicit dialogue planning to systematically structure interactions. In par-

allel, EMMA (Pantazopoulos et al., 2023) proposes a unified encoder-decoder framework for embodied task completion in Alexa Arena (Gao et al., 2023), formulating both action execution and dialogue grounding as a text generation problem. While EMMA achieves strong results through multitask training and dialogue-guided execution, it does not incorporate an explicit planning module to structure or optimise the clarification process across turns.

Finally, evaluating dialogue quality remains a challenge, particularly for coherence, informativeness, and engagement. Recent studies propose LLM-based evaluation techniques (Zhang et al., 2024; Abbasiantaeb et al., 2023), demonstrating that automated methods can provide reliable assessments of dialogue quality. Although they are efficient and scalable, a key issue emerges regarding the reliability of evaluation: the final results may be affected by errors originating from both the system and the evaluator.

Our work builds upon these foundations by explicitly incorporating dialogue planning into MLLMs. Inspired by prior work on Multi-Modal task execution (Hromei et al., 2024), we extend the scope beyond single-turn decision-making to multi-turn interactions, enabling systematic refinement of ambiguous commands through structured dialogue. This approach aligns with prior studies on collaborative dialogue in Minecraft (Narayan-Chen et al., 2019; Hromei et al., 2023b) but goes further by introducing an explicit planning mechanism to optimise the interaction process: a unified dialogue plan representation, similar in spirit to the structured interaction histories in (Padmakumar et al., 2021), where a multimodal transformer jointly models visual input and prior dialogue context for instruction-following. However, our method extends beyond reactive modelling by introducing an explicit, generative planning stage that precedes and guides the dialogue: the dynamic form of our approach allows the Agent to induce different plans (or re-plan) on the fly, support online search for alternative situations emerging *during the dialogue*.

3 A Multi-Modal Neural Architecture for Dialogue Planning in Grounded HRI

Prior approaches to Multi-Modal interaction in Minecraft-like environments, such as MM-IGLU (Hromei et al., 2024), only considered simple one-turn dialogues. The builder could ask only a single

clarification question before executing the command, and questions were drawn from a closed set with limited linguistic variability. This constraint often led to suboptimal task performance, as the system lacked mechanisms to iteratively refine its understanding of the user’s intent.

To overcome these limitations, we introduce a novel dialogical dataset tailored for Multi-Modal environments and propose a methodology for training Multi-Modal Language Models to handle more dynamic and adaptive conversations. Our approach defines structured information categories that the assistant should inquire about, enhancing its ability to formulate clarification questions systematically. Inspired by self-reflection (Huang et al., 2022) and chain-of-thought reasoning (Wei et al., 2022), our system first generates an information plan to guide the dialogue, identifying missing details, then iteratively refines its understanding by formulating targeted questions. Additionally, the agent must continuously reassess dialogue coherence, detecting inconsistencies and dynamically adjusting its strategy as needed. In the following, we present the Multi-Modal architectures employed in our approach, the underlying dialogue planning mechanism, and the methodology used to construct and train the dialogue models.

3.1 The MLLM Approach

Integrating visual and textual inputs is essential for an intelligent agent to make informed decisions based on both its environment and the commands it receives. To achieve this, Multi-Modal Large Language Models (MLLMs) employ different architectures to process and fuse image and text data effectively. Several approaches have been proposed, including LLaVA (Liu et al., 2024) and MiniCPM (Yao et al., 2024), each offering distinct trade-offs in terms of complexity and performance.

For instance, LLaVA (Liu et al., 2024) provides a straightforward integration of language and vision through a simple projector module, which transforms an image into the neural space of the language model, as a neural vector. For simplicity, we will call this vector s , as the scene depicted in the image. This design, built upon LLaMA (Touvron et al., 2023a,b; Dubey et al., 2024), enables efficient Multi-Modal learning while maintaining strong text-generation capabilities. On the other hand, MiniCPM (Yao et al., 2024) optimises the balance between parameter efficiency and performance. It employs a compression layer incorpo-

rating a perceiver resampler, similar to Flamingo (Alayrac et al., 2022), which utilises cross-attention to reduce the dimensionality of visual data while preserving critical features. The resulting compressed visual tokens s are then combined with textual inputs and processed by the underlying Qwen2-7B-based language model (Yang et al., 2024).

Formally, the agent’s response a is defined as a function MLLM acting over a dialogue input d and the reference scene s as an image in $\mathcal{I} : a = \text{MLLM}(d, s)$. These architectures process both the dialogue history and the reference image simultaneously, leveraging the attention mechanism of the language model to identify the most relevant contextual information. At run-time, the entire dialogue history is utilized to guide decision-making, ensuring that each response is grounded in the accumulated context of previous exchanges.

3.2 A Model for Dialogue Planning

Since the agent’s response depends on prior interactions, we need to define the *dialogue history* H_i at turn i as the sequence of past interactions (u_i, a_i) between the user and the agent, respectively. So, for $i = 2, \dots, k$, where k is the dialogue length, at a given i -th user’s turn we have:

$$\begin{aligned} H_i &= (u_i, a_{i-1}, H_{i-1}) & i > 1 \\ H_1 &= (u_1, \epsilon, H_\epsilon) \end{aligned} \quad (1)$$

whereas ϵ and H_ϵ correspond to the empty string and the empty triple, respectively, as no history is available at the first turn. As a consequence, the agent’s response a_i is therefore a function of both the dialogue history H_i and the scene s as an image in \mathcal{I} :

$$a_i = \text{MLLM}(\text{prompt}(H_i), s) \quad (2)$$

where prompt is a function used to instruct the model to reflect the history, i.e., the user turn u_i as well as the whole dialogue. Here, a_i corresponds to a confirmation about the command execution, i.e. as “Ok, I can do it” or a similar utterance, as a rejection of the command or as a clarification question when needed. The prompts adopted in this work are provided in the Appendix A.

As the history H_i makes explicit reference to the (u_i, a_{i-1}) pairs, the MLLM function is autoregressive. However, as the typical characteristic of standard LLMs, it presents significant drawbacks as the model must exhibit implicit reasoning capabilities, often leading to inconsistent responses,

redundant queries or even complete hallucinations. To mitigate these issues, we introduce the concept of *Dialogue Plan*: a structured and goal-driven sequence of clarification requests, i.e., a sequence of class labels in the set of abstract and general categories. These describe the different types of missing information, as discussed in the next Section, that guide the agent in solving any ambiguity in the requested task. In general, they are domain-agnostic, making the Dialogue Plan applicable to any scenario or domain while ensuring the clarification process becomes inherently Multi-Modal.

The Dialogue Plan organises the interaction step-by-step, guiding the agent to systematically address specific aspects of the user’s request. In some scenarios, the order of categories may be flexible, while in others, a fixed sequence is required to maintain logical coherence. For example, clarifying the orientation of an object might necessitate prior knowledge of its position or clarifying the departure time for a flight requires knowing the departure and arrival places. In this way, the Dialogue Plan enables the model to track what has already been asked, the responses received from the user, and the categories of information that still require clarification.

This explicit organisation reduces hallucinations by allowing the model to infer certain details from the context rather than redundantly querying the user. A dialogue history H_i can be augmented via a dialogue plan P , giving rise to the extended history H_i^+ , defined as follows:

$$\begin{aligned} H_i^+ &= (u_i, a_{i-1}, P_{i-1}, H_{i-1}^+) \quad i > 1 \\ H_1^+ &= (u_1, \epsilon, \epsilon, H_\epsilon^+) \end{aligned} \quad (3)$$

where no answer, no dialogue plan and no history are provided at the first step. The dialogue plan P_i is generated at each i -th turn by the same function MLLM , with the planning prompt $\text{prompt}(\cdot; p)$, defined as:

$$P_i = \text{MLLM}(\text{prompt}(H_i^+; p), s) \quad (4)$$

where the scene s is depicted in an image in \mathcal{I} . Notice how the planning ability exploits the same function MLLM also used as a language generation function. It is thus also used to produce optimal answers a_i over the scene s , coherent with the dialogue plan P_i , through the generation prompt $\text{prompt}(\cdot; g)$, as follows:

$$a_i = \text{MLLM}(\text{prompt}((H_i^+, P_i); g), s) \quad (5)$$

By structuring the interaction through a dialogue plan P_i , the dialogue supported by MLLM becomes more coherent and adaptive, by minimising inconsistencies and redundant queries. This method represents a modular and systematic approach to dialogue planning, improving the robustness and efficiency of Multi-Modal interactions.

3.3 Annotated Dialogue and MLLM Training

In our Minecraft-like environment, the plan dialogue P_i in the i -th turn of the dialogue is a sequence, whose members are types of missing information: for example, a category such as COLOR can be used to express that the colour of a block involved in a command is still unknown, and this limits the capability of the builder to carry out the underlying action. These types allow the agent to structure the dialogue effectively, as a specific question corresponds to an answer a_i at a given step of the plan. These categories correspond to a set $\mathcal{T} = \{t_1, \dots, t_n\}$ and allow the agent to systematically identify the still missing information to guide the dialogue in the clarification process. Every plan P_i is a sequence $(t_{i_1}, \dots, t_{i_n})$, where the missing type $t_{i_k} \in \mathcal{T}$ will be requested k steps after the i -th dialogue turn. The defined types $t_i \in \mathcal{T}$ used in this study and dependent on the Minecraft-like domain are summarised in Table 1. Intuitively, while the concept of dialogue planning is domain-agnostic, \mathcal{T} is domain-dependent, i.e., different domains require different instances of categories. As such, for adapting this methodology to a new domain, e.g., flight booking in air travel information systems, it is necessary to model the domain and define the $\mathcal{T}_{\text{Flights}}$ categories necessary for a typical dialogue.

The above types are the target of the agent’s

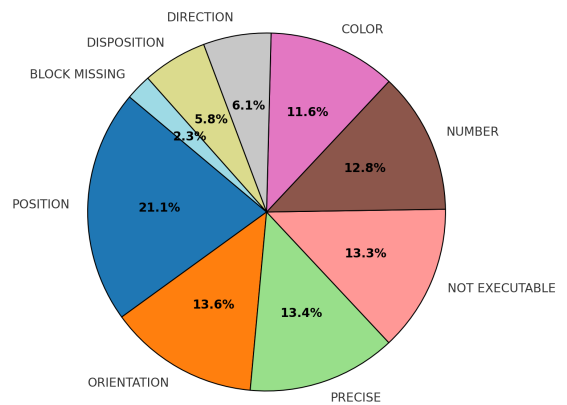


Figure 2: Category Distribution in the Annotated Dataset.

Category	Description
COLOR	Clarifies the colour of blocks when unspecified in the command.
NUMBER	Asks for the number of blocks required to fulfil the task.
ORIENTATION	Seeks the orientation of the builder (e.g., North, South, East, West).
DIRECTION	Determines the direction for placing a sequence of blocks.
BLOCK MISSING	Identifies cases where a specified block is absent on the map and requests clarification.
CONFIRMATION	Confirms that the command can be executed; includes a CONFIRMATION WITH RECAP subcategory that reiterates the command with a summary of the information gathered.
DISPOSITION	Inquires about the arrangement of blocks, with a subcategory PRECISE DISPOSITION for detailed alignment questions.
POSITION	Clarifies the location where blocks should be placed or removed from, with subcategories PRECISE POSITION and PRECISE BLOCK for detailed positioning or block identification.
NOT EXECUTABLE	Indicates when the command cannot be executed in cases of inconsistencies, with subcategories such as COMMAND, ACTION, and COLOR NOT FOUND for specific reasons.

Table 1: Dialogue clarification categories and their descriptions.

decision about the dialogue plan, i.e. the sequence $P_i = (t_{i_1}, \dots, t_{i_m})$ corresponding to the dynamic generation of clarification requests. This means that P_i is the sequence of turns used to produce the final answer a_m . In the example of Fig. 1, the u_1 command “Add some blocks above the easternmost purple column”, the agent will first produce the plan containing 5 types:

$$P_1 = (\text{NUMBER}, \text{COLOR}, \text{DISPOSITION}, \text{DIRECTION}, \text{CONFIRMATION})$$

They are necessary for gathering all the required information before accepting the executability of the action. In the example, $a_1 = \text{“How many blocks do I need to add above the easternmost purple column?”}$ is justified by the type $t_{1_1} = \text{NUMBER}$. On the subsequent turn, the agent will have to decide whether the user answered its question with relevant information, if its dialogue plan is still consistent with the newly received information and generate the new required dialogue plan:

$$P_2 = (\text{COLOR}, \text{DISPOSITION}, \text{DIRECTION}, \text{CONFIRMATION})$$

In this way, at each $(i + 1)$ -th interaction, the agent will exploit type t_{i_1} to produce the next relevant question a_{i+1} , in the case of Figure 1 “What colour should the blocks be?”.

The distribution of these adopted missing information types in the plans of our dialogue dataset is reported in Figure 2. It can be observed that they are almost evenly distributed, with a few examples of BLOCK MISSING. Notice that the CONFIRMATION is not reported, since all our dialogues end with a confirmation from the Builder, since our goal was to provide dialogues where, at some point, all ambiguities are resolved!

The structured planning mechanism was used to annotate a dataset of 1,100 dialogues in English and Italian, created by five annotators based on MM-IGLU-generated images (Hromei et al., 2024). The dialogues were designed to simulate real-world scenarios, including ambiguous or misleading commands, to test model robustness. Annotators freely generated interactions starting from a given scene s , defining an initial command u_1 , the Builder’s plan p_i , clarification questions a_i , and user responses u_i , iterating until task completion. Dialogues average seven turns, ranging from simple two-turn exchanges to complex interactions of up to twenty turns. The Builder refines its understanding iteratively, handling diverse tasks such as shape construction, block removal, and figure assembly.

4 Experimental Evaluation

Our evaluation examines the impact of explicit dialogue planning in Multi-Modal LLMs by addressing three key aspects: *i*) whether fine-tuning on a dedicated dataset improves performance over zero-shot inference, *ii*) how the dialogue planning phase enhances dialogue efficiency at both the turn and full dialogue level, and *iii*) whether the dialogue planning phase facilitates cross-lingual generalisation. To assess the effects of fine-tuning, we compare zero-shot performance against models fine-tuned on our annotated dataset. For dialogue efficiency, we contrast models trained solely on dialogue history with those incorporating structured dialogue planning, measuring improvements in response quality, reduction of redundant queries, and overall interaction effectiveness.

The dataset (see Table 2) consists of 1,136 manually annotated dialogues, with an average of 7.52 turns per dialogue, and is split into training (908), development (113), and test (115) sets following

	Lang	Train	Dev	Test
Dialogues Number	-	908	113	115
AVG Turns Number	-	7,52	7,40	7,60
AVG Turn Lenght	EN	10,71	9,77	10,37
	IT	9,77	8,91	9,56
AVG Recap Lenght	EN	23,71	21,30	23,09
	IT	22,08	19,87	22,19

Table 2: Dataset statistics: number of dialogues, average turns, and turn/recap lengths in English and Italian, split for training, development, and testing.

an 80-10-10 ratio. Each dialogue is available in both English and Italian to enable controlled cross-lingual evaluations. The inclusion of a manually translated Italian version ensures that cross-lingual assessments measure generalisation effects rather than language-specific variations. We assess whether planning enhances cross-lingual robustness by evaluating model transfer between English and Italian, testing if structured planning improves performance across languages.

Turn-Driven Evaluation. This evaluation focuses on the model’s ability to generate responses at the individual turn level. The agent (LLM) must determine whether additional clarification is needed before execution. If necessary, it generates a clarification question (*Q*); otherwise, it provides a recap (*RC*) summarising its understanding of the task. To assess these aspects, we evaluate clarification questions and recap generation separately.

To examine the impact of fine-tuning, we test two state-of-the-art Multi-Modal models with a compact number of parameters: LLaVA (Liu et al., 2024) and MiniCPM (Yao et al., 2024). These models are evaluated in a zero-shot setting, where they rely solely on a system prompt, and in a fine-tuned setting, where they learn to generate structured plans before responding. For plan generation, the model receives a context consisting of the initial task-defining prompt (see Appendix A), the dialogue history up to that turn, and an image. We assess the alignment of generated plans with expected outputs using Precision, Recall, and F1-score for the plan categories defined in Section 3. For clarification questions and recap generation, output quality is measured using cosine similarity between the embedding vectors of generated responses and expected outputs, to better capture the semantic and pragmatic quality of responses in our highly contextual, interactive HRI scenario: notice that rather than simple text generation (as in translation), we focus on suitable linguistic interactions

as capable to preserve all inferences needed by the underlying HRI tasks (e.g. posing questions such as potential paraphrases). This capability is very weakly modelled by machine translation metrics such as BLEU or ROUGE. This is computed using the Sentence-BERT model (Reimers and Gurevych, 2019)². Higher cosine similarity indicates greater syntactic and semantic alignment with the expected clarification and recap.

Fine-tuning was conducted on four A100 GPUs using a total batch size of 8 and a learning rate of $5 \cdot 10^{-5}$. Complexity reduction techniques such as LoRA (rank R and α set to 64) were applied to reduce training costs in terms of time and energy, to the attention and output layers of the language decoding module. Moreover, we continued the joint fine-tuning of the vision encoder as well, as detailed in the MiniCPM paper (Yao et al., 2024).

Setting	Model	Plan			Generation	
		P	R	F1	Q	RC
zero-shot	LLaVA	-	-	-	.37	.39
	MiniCPM	-	-	-	.55	.71
Fine-tuned (with Plan)	LLaVA	.93	.90	.92	.82	.91
	MiniCPM	.90	.91	.91	.84	.96

Table 3: Comparison of LLaVA and MiniCPM in zero-shot and fine-tuned settings for plan accuracy and response generation.

Table 3 compares LLaVA and MiniCPM across both evaluation settings, highlighting the impact of fine-tuning with structured planning. The results show that fine-tuned models effectively classify planning categories, achieving over 90% F1-score in plan generation. In contrast, baseline zero-shot models struggle to generate contextually appropriate responses, with LLaVA scoring only 0.37 in the similarity between its clarification questions and expected ones. Fine-tuning significantly improves performance, with LLaVA reaching 0.82 when incorporating planning. Comparing the two models, fine-tuned MiniCPM outperforms LLaVA in question and recap generation, while both models achieve similar performance in planning.

To further assess the impact of explicit planning beyond fine-tuning alone, we conduct a detailed evaluation comparing MiniCPM in settings with and without structured planning. Given its consistently superior performance across previous experiments, we focus exclusively on MiniCPM for this analysis. Table 4 presents results across three con-

²<https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2>

Setting	Language		Generation	
	Train	Test	Question	Recap
zero-shot	-	EN	0.548	0.706
	-	IT	0.595	0.727
Fine-tuned (no Plan)	EN	EN	0.768	0.921
	EN	IT	0.659	0.861
	IT	IT	0.794	0.925
	IT	EN	0.638	0.908
Fine-tuned (with Plan)	EN	EN	0.843	0.955
	EN	IT	0.735	0.930
	IT	IT	0.867	0.947
	IT	EN	0.738	0.927

Table 4: MiniCPM performance in zero-shot, fine-tuned, and planning-enhanced settings across monolingual and cross-lingual evaluations.

figurations: zero-shot, fine-tuning on dialogue history alone, and fine-tuning with explicit planning. In the monolingual EN→EN scenario, fine-tuning without planning significantly improves generation quality compared to the zero-shot model (e.g., Question Generation rises from 0.548 to 0.768, while Recap Generation increases from 0.706 to 0.921). However, the addition of planning further boosts performance, with Question Generation reaching 0.843 and Recap Generation improving to 0.955.

These findings confirm that fine-tuning alone enhances response quality, but planning refines the interaction by making clarification requests more structured and summaries more accurate. We also examine cross-lingual generalisation, where the model is trained in one language and tested in another. The results reveal that models fine-tuned only on dialogue history experience a notable drop in Question Generation accuracy when applied across languages (EN→IT: 0.659, IT→EN: 0.638). This degradation suggests that, without structured planning, the model struggles to transfer the reasoning process necessary for effective question formulation. In contrast, incorporating planning significantly mitigates this issue: the EN→IT model achieves 0.735, and the IT→EN model reaches 0.738, both exceeding the performance of the history-only model in monolingual settings.

Interestingly, Recap Generation remains more stable across settings, with smaller differences between monolingual and cross-lingual results. This indicates that summarisation is a more transferable skill, whereas effective clarification question generation benefits more directly from structured planning. These findings underscore the role of explicit planning not only in improving dialogue

quality but also in strengthening the model’s ability to generalise across languages, reinforcing the importance of structured Multi-Modal training for robust cross-lingual HRI applications.

Dialogue Evaluation. In the previous evaluation, we analysed the performance of the model at the individual response level, leveraging the gold-standard dialogue history to assess its ability to generate appropriate clarification questions and recaps. However, in this phase, we shift our focus to evaluating the overall quality of full dialogues rather than isolated responses. Specifically, we aim to answer the question: *How well can the model sustain a complete interaction that dynamically adapts to user input while effectively gathering the necessary information?* To achieve this, we deploy the fine-tuned MiniCPM model in an interactive setting, allowing it to generate dialogues in real-time.

Two independent annotators, who had not participated in the dataset annotation, conducted this evaluation. Prior to the study, they were given only a brief introduction: they reviewed ten training examples and read descriptions of the dialogue categories to familiarise themselves with the interaction structure. Each annotator randomly selected a test case, viewed the corresponding world image, and initialised the system by providing the user’s first request, after reading the gold-standard recap to understand the intended task outcome. From that point, they interacted with the model by answering its clarification questions, ensuring they provided the necessary details for the system to achieve the expected goal. This process was repeated across 40 dialogues, enabling a comprehensive evaluation of how effectively the model conducts a full conversation and adapts to evolving user input.

To assess how efficiently the model conducts interactions, we measure the Dialogue Efficiency Cost, defined as the ratio between the actual number of turns and the ideal number needed to complete the task. A lower score indicates that the model gathers necessary information with fewer exchanges, highlighting the impact of planning on dialogue effectiveness.

At the end of each interaction, annotators completed a brief questionnaire to assess the overall quality of the dialogue. Specifically, they provided ratings on two key dimensions: relevance and fluency. Relevance measures the model’s ability to seek and provide only necessary information, ensur-

ing that each turn contributes meaningfully to task completion without unnecessary or redundant exchanges. A higher relevance score indicates that the dialogue remains focused and aligned with user intent. Fluency assesses how naturally and coherently the assistant formulates questions and responses, prioritising clarity, grammatical correctness, and ease of comprehension. Poor fluency may result in unnatural phrasing or disjointed interactions, while high fluency ensures smooth and user-friendly communication. Annotators rated these dimensions on a five-point scale, from 1 (EXTREMELY POOR) to 5 (EXCELLENT). The full questionnaire and scoring details are provided in Appendix C.

Setting	AVG Length	Dialogue Efficiency	Relevance	Fluency
no Plan	12.5	1.05	3.13	4.88
with Plan	11.6	0.98	4.25	4.90

Table 5: Dialogue evaluation results

Table 5 presents a comparative analysis of model performance with and without planning across four key metrics: dialogue efficiency, relevance, fluency, and average dialogue length. The results show that incorporating a planning phase leads to notable improvements in dialogue quality. In particular, the planned model achieves a significantly higher relevance score (4.25 vs. 3.13), suggesting that it generates more contextually appropriate and purposeful exchanges, avoiding unnecessary, off-topic questions. Similarly, fluency improves slightly (4.90 vs. 4.88), indicating that while both settings produce grammatically well-formed responses, the planned model better maintains conversational coherence.

From an efficiency perspective, the model with planning achieves a dialogue efficiency score closer to the optimal value of 1 (0.98 vs. 1.05), suggesting that it requires fewer redundant turns to reach the intended task completion. This is further reflected in the average dialogue length, where the planned model produces slightly shorter interactions (11.6 vs. 12.5), potentially indicating a more structured and goal-driven exchange. These results indicate that explicit planning improves dialogue relevance while offering a moderate gain in efficiency by keeping interactions more focused and reducing unnecessary exchanges. An error analysis is shown in Appendix B.

5 Conclusion

This work explored the integration of explicit dialogue planning in Multi-Modal Large Language Models for Human-Robot Interaction. We proposed a structured planning mechanism to enhance an agent’s ability to gather task-relevant information through dialogue before executing user commands. To support this investigation, we introduced a dataset of over 1,100 annotated dialogues in English and Italian, enabling fine-tuning and cross-lingual evaluation.

Our experimental results demonstrate that fine-tuning significantly improves performance over zero-shot inference, while explicit planning further enhances response accuracy, reduces redundant exchanges, and improves overall dialogue efficiency. Additionally, we show that planning facilitates cross-lingual generalisation, allowing models trained in one language to transfer effectively to another. In this way, we contribute a novel benchmark and methodology to advance future research on grounded, multi-modal dialogue systems for HRI.

Future directions include investigating alternative planning strategies, such as integrating planning into a single-step generation process, exploring more advanced learning paradigms like reinforcement learning, and extending this approach to less structured environments with dynamically evolving configurations.

Acknowledgments

We acknowledge financial support from the PNRR project FAIR - Future AI Research (PE00000013), under the NRRP MUR program funded by the NextGenerationEU and support from Project ECS 0000024 Rome Technopole - CUP B83C22002820006, NRP Mission 4 Component 2 Investment 1.5, Funded by the European Union - NextGenerationEU. Claudiu Daniel Hromei is a Ph.D. student enrolled in the National Ph.D. in Artificial Intelligence, XXXVII cycle, course on *Health and life sciences*, organized by the Università Campus Bio-Medico di Roma.

Limitations

While our approach demonstrates the benefits of explicit dialogue planning in Multi-Modal Large Language Models (MLLMs) for Human-Robot Interaction (HRI), several limitations remain.

First, the proposed planning mechanism relies on predefined clarification categories, which may not generalise to all HRI scenarios. Although these categories improve dialogue efficiency, they impose a structured framework that might limit the flexibility needed for open-ended or highly dynamic interactions.

Second, our dataset consists of dialogues set in a block-based, synthetic environment with static world configurations. While this controlled setting allows for systematic evaluation, real-world HRI scenarios often involve environments that change over time and require continuous adaptation. Extending this approach to more complex, evolving contexts would be a natural next step.

Third, the planning step is currently treated as a separate generation process. While this modular approach facilitates structured reasoning, it may introduce additional latency. Future work could explore integrating planning directly into the response generation step, potentially reducing inference time while maintaining structured dialogue management.

Finally, our experiments focus on fine-tuning MLLMs, but alternative training paradigms, such as reinforcement learning, could further refine planning strategies and enable models to adapt dynamically based on user interactions. Investigating such approaches could improve long-term engagement and generalisation in interactive systems.

References

- Zahra Abbasiantaeb, Yifei Yuan, Evangelos Kanoulas, and Mohammad Aliannejadi. 2023. [Let the llms talk: Simulating human-to-human conversational qa via zero-shot llm-to-llm interactions](#). *Preprint*, arXiv:2312.02913.
- Seyed Hossein Alavi, Sudha Rao, Ashutosh Adhikari, Gabriel A DesGarennas, Akanksha Malhotra, Chris Brockett, Mahmoud Adada, Raymond T. Ng, Vered Schwartz, and Bill Dolan. 2024. [Mcpdial: A minecraft persona-driven dialogue dataset](#). *Preprint*, arXiv:2410.21627.
- Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob Menick, Sebastian Borgeaud, Andrew Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karen Simonyan. 2022. [Flamingo: a visual language model for few-shot learning](#). *Preprint*, arXiv:2204.14198.
- Giuseppe Attanasio, Pierpaolo Basile, Federico Borazio, Danilo Croce, Maria Francis, Jacopo Gili, Elio Musacchio, Malvina Nissim, Viviana Patti, Matteo Rinaldi, and Daniel Scalena. 2024. [CALAMITA: Challenge the abilities of LAnguage models in ITALian](#). In *Proceedings of the 10th Italian Conference on Computational Linguistics (CLiC-it 2024)*, Pisa, Italy. CEUR Workshop Proceedings.
- Zechen Bai, Pichao Wang, Tianjun Xiao, Tong He, Zongbo Han, Zheng Zhang, and Mike Zheng Shou. 2024. [Hallucination of multimodal large language models: A survey](#). *Preprint*, arXiv:2404.18930.
- Federico Borazio, Danilo Croce, Giorgio Gambosi, Roberto Basili, Daniele Margiotta, Antonio Scaiella, Martina Del Manso, Daniele Petrone, Andrea Cannone, Alberto M. Urdiales, Chiara Sacco, Patrizio Pezzotti, Flavia Riccardo, Daniele Mipatrini, Federica Ferraro, and Sobha Pilati. 2024. [Semi-automatic topic discovery and classification for epidemic intelligence via large language models](#). In *Proceedings of the Second Workshop on Natural Language Processing for Political Sciences @ LREC-COLING 2024*, pages 68–84, Torino, Italia. ELRA and ICCL.
- Federico Borazio, Claudiu Daniel Hromei, Elisa Pasone, Danilo Croce, and Roberto Basili. 2025. [Mm-iglu-it: Multi-modal interactive grounded language understanding in italian](#). In *AIxIA 2024 – Advances in Artificial Intelligence*, pages 64–78, Cham. Springer Nature Switzerland.
- Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.
- Qiaozi Gao, Govind Thattai, Xiaofeng Gao, Suhaila Shakiah, Shreyas Pansare, Vasu Sharma, Gaurav Sukhatme, Hangjie Shi, Bofei Yang, Desheng Zheng, et al. 2023. [Alexa arena: A user-centric interactive platform for embodied ai](#). *arXiv preprint arXiv:2303.01586*.
- Xiaofeng Gao, Qiaozi Gao, Ran Gong, Kaixiang Lin, Govind Thattai, and Gaurav S. Sukhatme. 2022. [Dialfred: Dialogue-enabled agents for embodied instruction following](#). *IEEE Robotics and Automation Letters*, 7(4):10049–10056.
- Elliot Gestrin, Marco Kuhlmann, and Jendrik Seipp. 2024. [NL2plan: Robust llm-driven planning from minimal text descriptions](#). *Preprint*, arXiv:2405.04215.
- Shibo Hao, Yi Gu, Haodi Ma, Joshua Jiahua Hong, Zhen Wang, Daisy Zhe Wang, and Zhiting Hu. 2023. [Reasoning with language model is planning with world model](#). *Preprint*, arXiv:2305.14992.
- Claudiu Daniel Hromei, Lorenzo Cristofori, Danilo Croce, and Roberto Basili. 2022a. [Embedding contextual information in seq2seq models for grounded semantic role labeling](#). In *AIxIA 2022 - Advances*

- in *Artificial Intelligence - XX1st International Conference of the Italian Association for Artificial Intelligence, AIxIA 2022, Udine, Italy, November 28 - December 2, 2022, Proceedings*, volume 13796 of *Lecture Notes in Computer Science*, pages 472–485. Springer.
- Claudiu Daniel Hromei, Danilo Croce, and Roberto Basili. 2022b. [Grounding end-to-end architectures for semantic role labeling in human robot interaction](#). In *Proceedings of the Sixth Workshop on Natural Language for Artificial Intelligence (NL4AI 2022) co-located with 21th International Conference of the Italian Association for Artificial Intelligence (AI*IA 2022)*, Udine, November 30th, 2022, volume 3287 of *CEUR Workshop Proceedings*, pages 24–38. CEUR-WS.org.
- Claudiu Daniel Hromei, Danilo Croce, and Roberto Basili. 2023a. [Grounding end-to-end pre-trained architectures for semantic role labeling in multiple languages](#). *Intelligenza Artificiale*, 17(2):173–191.
- Claudiu Daniel Hromei, Daniele Margiotta, Danilo Croce, and Roberto Basili. 2023b. [An end-to-end transformer-based model for interactive grounded language understanding](#). In *Proceedings of the Seventh Workshop on Natural Language for Artificial Intelligence (NL4AI 2023) co-located with 22th International Conference of the Italian Association for Artificial Intelligence (AIxIA 2023)*, Rome, Italy, November 6th-7th, 2023, volume 3551 of *CEUR Workshop Proceedings*. CEUR-WS.org.
- Claudiu Daniel Hromei, Daniele Margiotta, Danilo Croce, and Roberto Basili. 2024. [MM-IGLU: Multimodal interactive grounded language understanding](#). In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 11440–11451, Torino, Italia. ELRA and ICCL.
- Chenguang Huang, Oier Mees, Andy Zeng, and Wolfram Burgard. 2023. [Visual language maps for robot navigation](#). *Preprint*, arXiv:2210.05714.
- Jiaxin Huang, Shixiang Shane Gu, Le Hou, Yuexin Wu, Xuezhi Wang, Hongkun Yu, and Jiawei Han. 2022. Large language models can self-improve. *arXiv preprint arXiv:2210.11610*.
- Xue Jiang, Yihong Dong, Lecheng Wang, Zheng Fang, Qiwei Shang, Ge Li, Zhi Jin, and Wenpin Jiao. 2024. [Self-planning code generation with large language models](#). *ACM Trans. Softw. Eng. Methodol.*, 33(7).
- Julia Kiseleva, Alexey Skrynnik, Artem Zholus, Shrestha Mohanty, Negar Arabzadeh, Marc-Alexandre Côté, Mohammad Aliannejadi, Milagro Teruel, Ziming Li, Mikhail Burtsev, Maartje ter Hove, Zoya Volovikova, Aleksandr Panov, Yuxuan Sun, Kavya Srinet, Arthur Szlam, and Ahmed Awadallah. 2022. [Iglu 2022: Interactive grounded language understanding in a collaborative environment at neurips 2022](#). *Preprint*, arXiv:2205.13771.
- Kunze Li and Yu Zhang. 2024. [Planning first, question second: An LLM-guided method for controllable question generation](#). In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 4715–4729, Bangkok, Thailand. Association for Computational Linguistics.
- Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2024. Visual instruction tuning. *Advances in neural information processing systems*, 36.
- Anjali Narayan-Chen, Prashant Jayannavar, and Julia Hockenmaier. 2019. [Collaborative dialogue in Minecraft](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5405–5415, Florence, Italy. Association for Computational Linguistics.
- Aishwarya Padmakumar, Jesse Thomason, Ayush Shrivastava, Patrick Lange, Anjali Narayan-Chen, Spandana Gella, Robinson Piramuthu, Gokhan Tur, and Dilek Hakkani-Tur. 2021. [Teach: Task-driven embodied agents that chat](#). *Preprint*, arXiv:2110.00534.
- Georgios Pantazopoulos, Malvina Nikandrou, Amit Parekh, Bhathiya Hemanthage, Arash Eshghi, Ioannis Konstantas, Verena Rieser, Oliver Lemon, and Alessandro Suglia. 2023. [Multitask multimodal prompted training for interactive embodied task completion](#). *Preprint*, arXiv:2311.04067.
- Nils Reimers and Iryna Gurevych. 2019. [Sentence-bert: Sentence embeddings using siamese bert-networks](#). *Preprint*, arXiv:1908.10084.
- Antonio Scaiella, Daniele Margiotta, Claudiu Daniel Hromei, Danilo Croce, and Roberto Basili. 2024. Evaluating multimodal large language models for visual question-answering in italian. In *Proceedings of the Eighth Workshop on Natural Language for Artificial Intelligence (NL4AI 2024) co-located with 23th International Conference of the Italian Association for Artificial Intelligence (AIxIA 2024)*, Bolzano, Italy, November 25th-28th, 2024, CEUR Workshop Proceedings. CEUR-WS.org.
- Mohit Shridhar, Jesse Thomason, Daniel Gordon, Yonatan Bisk, Winson Han, Roozbeh Mottaghi, Luke Zettlemoyer, and Dieter Fox. 2020. [Alfred: A benchmark for interpreting grounded instructions for everyday tasks](#). *Preprint*, arXiv:1912.01734.
- Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023a. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.
- Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shrutika Bhosale, et al. 2023b. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. *arXiv preprint arXiv:2201.11903*.

Zhenyu Wu, Ziwei Wang, Xiuwei Xu, Jiwen Lu, and Haibin Yan. 2023. [Embodied task planning with large language models](#). *Preprint*, arXiv:2307.01848.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zhihao Fan. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

Yuan Yao, Tianyu Yu, Ao Zhang, Chongyi Wang, Junbo Cui, Hongji Zhu, Tianchi Cai, Haoyu Li, Weilin Zhao, Zhihui He, et al. 2024. Minicpm-v: A gpt-4v level mllm on your phone. *arXiv preprint arXiv:2408.01800*.

Shukang Yin, Chaoyou Fu, Sirui Zhao, Ke Li, Xing Sun, Tong Xu, and Enhong Chen. 2024. [A survey on multimodal large language models](#). *National Science Review*, 11(12):nwae403.

Bo Zhang, Hui Ma, Jian Ding, Jian Wang, Bo Xu, and Hongfei Lin. 2025. [Distilling implicit multimodal knowledge into large language models for zero-resource dialogue generation](#). *Information Fusion*, page 102985.

Chen Zhang, Luis Fernando D’Haro, Yiming Chen, Malu Zhang, and Haizhou Li. 2024. [A comprehensive analysis of the effectiveness of large language models as automatic dialogue evaluators](#). *Preprint*, arXiv:2312.15407.

A Prompt Design

In this section, we describe the prompts used to condition the MiniCPM³ and LLaVA⁴ models for the experimental analysis (Section 4), both in English and Italian. The prompts are categorised into two primary settings:

- **Without Plan Setting:** In this configuration, the model does not rely on a predefined plan. The prompt instructs the model to either:
 - Generate a clarification question based on the given dialogue history, reference image, and user command. The model must identify missing or ambiguous information related to specific categories such as block colour, number, orientation, direction, position, or the feasibility of actions and commands.
 - Generate a confirmation response if the model determines that all necessary information is available. This response can either affirm the execution of the command or summarise all prior executed actions relevant to the user request.

This type of prompt has been used in both the **zero-shot** setting and the **fine-tuned (no Plan)** scenario. See Figures 7 and 8 for reference.

- **With Plan Setting:** In this setting, the model is conditioned to first generate a structured plan before producing a question or confirmation. This enables a more systematic approach to gathering necessary information. The setting includes two distinct prompt types:
 - **Plan Generation Prompt:** Given the dialogue history, user command, and reference image, the model is instructed to generate a structured *plan*, listing the categories of missing information that require clarification. The categories are predefined, covering aspects such as block colour, number, spatial arrangement, or command feasibility. The model must ensure consistency with previously generated plans when applicable. See Figures 9 and 10.

³https://huggingface.co/openbmb/MiniCPM-V-2_6

⁴<https://huggingface.co/meta-llama/llama-2-7b-chat-hf>

- **Clarification/Confirmation Prompt:** Once a plan is available, the model is instructed to generate either: *i*) a clarification question corresponding to the first category in the input plan; *ii*) a confirmation response if the plan indicates that all necessary information is available. This response may include an explicit recap of the actions performed. See Figures 11 and 12.

Each prompt follows a structured format:

- **Context Definition:** The model is provided with structured input, including a world image, a history of prior interactions, and (if applicable) a previously generated plan.
- **Task Objective:** The model is explicitly instructed to either generate a clarification question, formulate a confirmation, or produce a structured plan, depending on the setting.
- **Categorical Breakdown:** The prompts define the missing information through the specific categories in \mathcal{T} , ensuring that the model systematically assesses what aspects require clarification before executing a command. The set \mathcal{T} includes the following categories:
 - **COLOR:** Clarifications about block color.
 - **NUMBER:** Clarifications about the number of blocks.
 - **ORIENTATION:** Direction in which the model should operate.
 - **DIRECTION:** Direction of block placement.
 - **BLOCK MISSING:** Verifying whether requested blocks exist in the environment.
 - **NOT EXECUTABLE ACTION:** Determining whether an action can be performed.
 - **NOT EXECUTABLE COMMAND:** Determining whether a command can be executed.
 - **NOT EXECUTABLE: COLOR NOT FOUND:** Addressing unavailable block colors.
 - **DISPOSITION:** General arrangement of blocks.
 - **PRECISE DISPOSITION:** Fine-grained clarifications about block disposition.
 - **POSITION:** General clarification about block positioning.

- **PRECISE POSITION:** Detailed clarification about block positioning.
- **PRECISE BLOCK:** Clarifications about a specific block in question.
- **CONFIRMATION:** Affirming the execution of a command.
- **CONFIRMATION WITH RECAP:** Summarizing all previous actions relevant to the command.

- **Task Execution:** Finally, the model is instructed to generate a well-formed response that adheres to the specified categories (either posing a clarification question), confirming the execution of the command, or summarizing prior actions.

B Error Analysis

Figure 3 presents three illustrative scenarios from the dynamic dialogue test set, where a human evaluator independently interacts with both models, one incorporating planning and the other operating without it. Below, we summarise key observations.

In the first scenario (Figure 3a), the user provides a vague command, “*Place some blocks*”. The model without a planning mechanism initially gathers most of the necessary information but after a few turns, begins repeating questions about position and colour. Moreover, it fails to ask about the arrangement of the blocks (e.g., whether they should be placed in a line, column, or scattered). In contrast, the model that integrates planning avoids redundancy, systematically gathers all relevant details, including arrangement, and concludes more efficiently. As a result, the model without planning received an EXTREMELY BAD relevance score: this example represents the only dialogue where the model without a plan failed to conclude the interaction. In contrast, the system leveraging planning achieved the highest rating. We report the entire dialogue in Figure 4.

In the second scenario (Figure 3b), the user instructs, “*Add some black blocks and then add four blocks of another color.*” Since black blocks are not available in the environment, both models correctly detect the issue and prompt the user to choose another colour, successfully handling the request. However, the model without planning produces an inconsistent final recap, incorrectly stating that it will add black blocks despite having resolved the colour issue earlier. This inconsistency highlights a

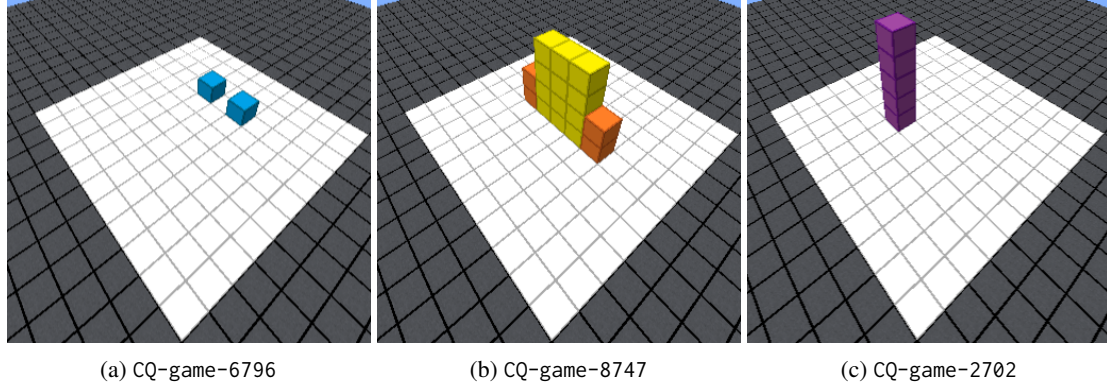


Figure 3: Environments examples of some dialogues we tested the models on.

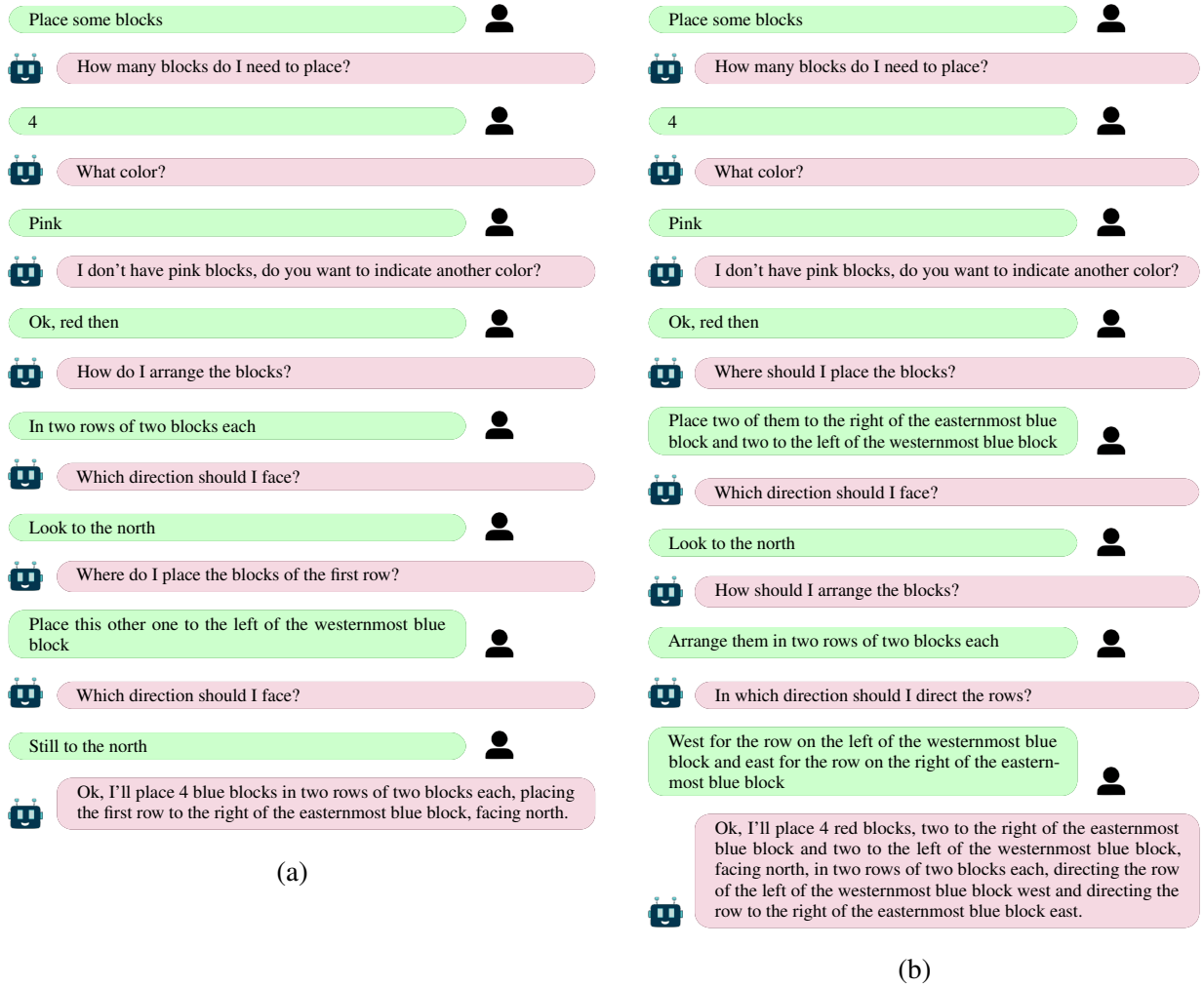


Figure 4: Dialogues related to the CQ-game-6796 image from Figure 3a. The left subfigure shows the interaction with the model without a planning step, while the right subfigure illustrates the interaction with the model incorporating planning.

failure to maintain coherence. The model equipped with planning, on the other hand, avoids this issue, maintaining logical consistency throughout the dialogue. We report the entire dialogue in Figure 5.

The third scenario (Figure 3c) is associated with

“Add blocks to create the number ten on the map”. The model, lacking planning, misinterprets the request, asking irrelevant questions such as how many blocks are needed and incorrectly assuming the task involves creating a column. The model using planning correctly, infers the number of blocks

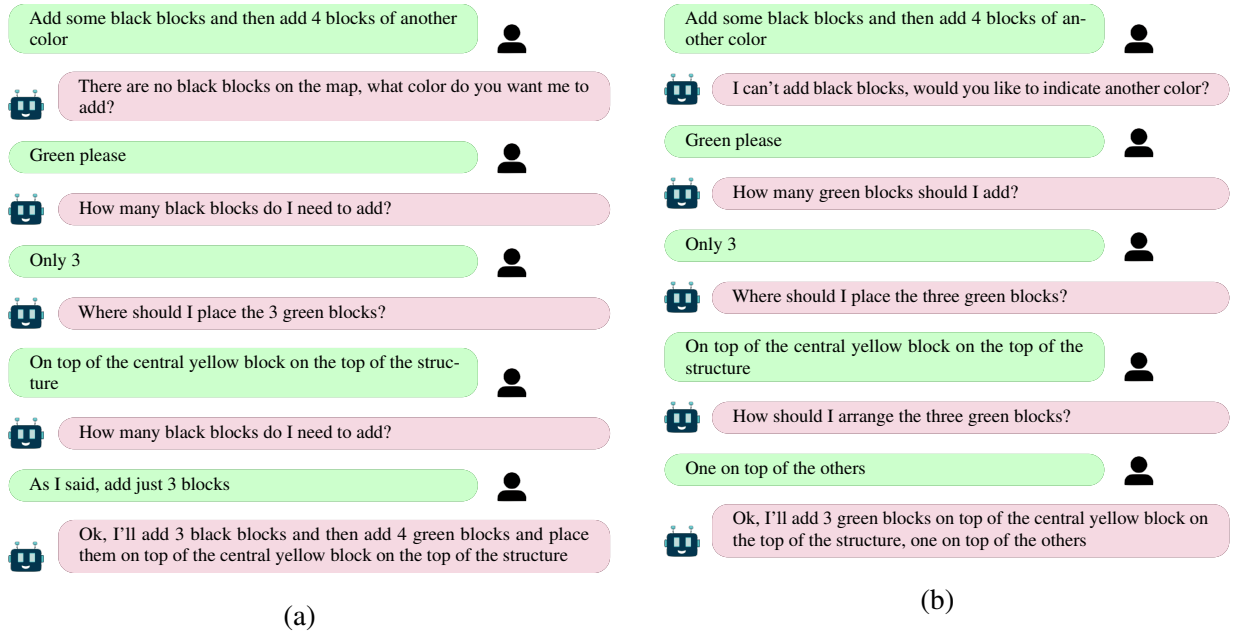


Figure 5: Dialogues related to the CQ-game-8747 image from Figure 3b. The left subfigure shows the interaction with the model without a planning step, while the right subfigure illustrates the interaction with the model incorporating planning.

from the image but does not explicitly confirm this information in its final response. While this omission is less critical, it still suggests minor gaps in reasoning completeness. We report the entire dialogue in Figure 6.

C Evaluation Questionnaire

As discussed in Section 4, to conduct a comprehensive analysis of the dialogue quality, we integrated human evaluation into our framework. Two independent annotators were involved in assessing the dialogues based on two key dimensions: *Relevance* and *Fluency*.

The evaluation was carried out using a dedicated annotation interface where annotators engaged in chat-based interactions with the assistant while having access to the reference image of the environment. Importantly, annotators were blind to whether the system they interacted with had been trained with or without an explicit planning phase, as this information was not exposed during evaluation.

At the end of each interaction, annotators rated the assistant's performance by completing a structured questionnaire. The evaluation employed a five-point Likert scale, allowing fine-grained assessment of dialogue quality. The criteria for assigning scores are outlined in Figure 13.

The *Relevance* dimension evaluates whether the

assistant asks only necessary clarification questions, avoiding irrelevant or redundant requests. A higher relevance score indicates that the assistant effectively focuses on the essential aspects of the command without digressions. At the lowest end of the scale, the assistant fails to request crucial clarifications or asks entirely irrelevant questions. As the score increases, the assistant progressively refines its queries, reducing unnecessary questions while maintaining the essential ones. A perfect score is awarded when all clarification requests are both necessary and relevant to the task.

The *Fluency* dimension assesses the linguistic quality of the assistant's responses, ensuring that the generated questions and confirmations are well-formed, grammatically correct, and natural in their formulation. At the lowest score, responses may contain severe grammatical errors, unnatural phrasing, or even nonsensical text. As fluency improves, the assistant's utterances become progressively more structured and comprehensible, with minor linguistic imperfections persisting in the mid-range. A maximum score is given when the language is completely natural, error-free, and appropriately structured for a seamless interaction.

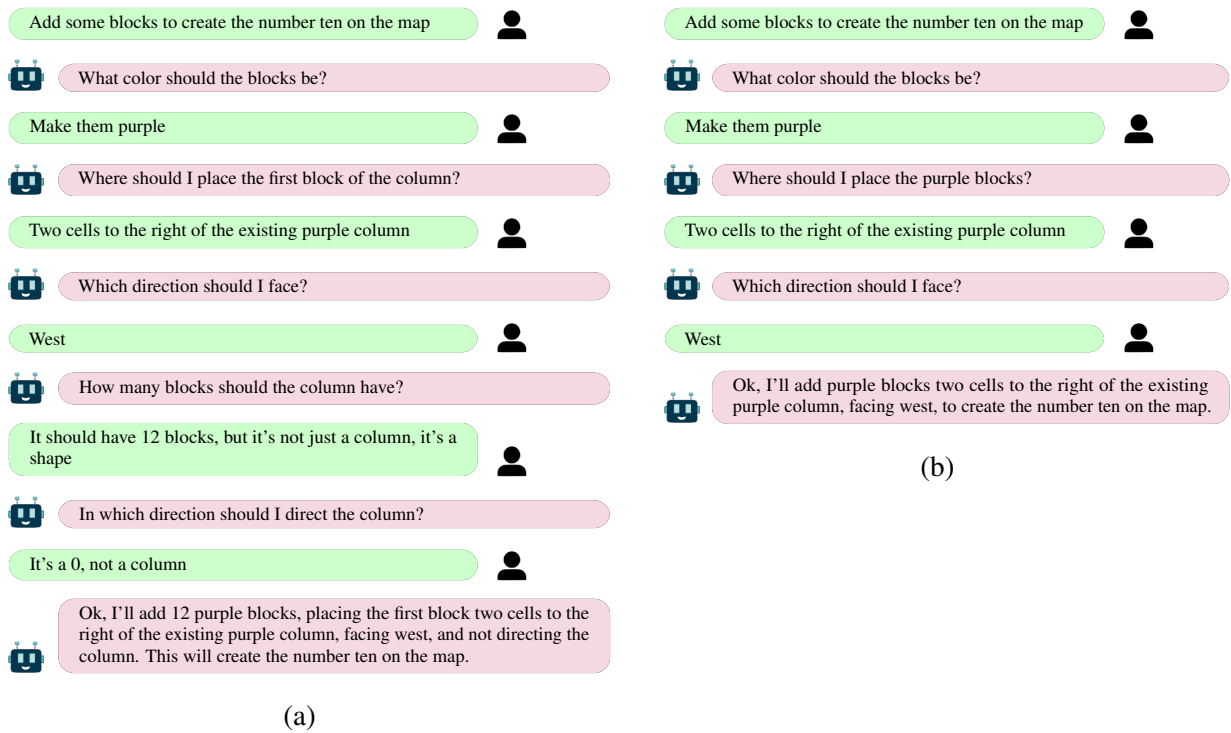


Figure 6: Dialogues related to the CQ-game-2702 image from Figure 3c. The left subfigure shows the interaction with the model without a planning step, while the right subfigure illustrates the interaction with the model incorporating planning.

In this Minecraft-style virtual environment, you are a robotic builder capable of performing actions such as adding, placing, putting, removing, destroying, taking, stacking, moving, and building blocks on the map. Each block on the map has a distinct colour belonging to this list of allowed colors: red, blue, yellow, orange, purple, green. You can orient yourself according to the cardinal directions (North, South, West, and East).

Your task is: given the user's command, the map image, and the possible history of the dialogue (if present), you must generate a clarification question as output. The possible categories for generating clarification questions are as follows:

- block colors
- number of blocks
- the orientation you need to have to perform an operation
- the direction of blocks
- blocks not present on the map
- actions that cannot be performed
- commands that cannot be performed
- colors that cannot be used
- block arrangement
- precise arrangement of blocks
- block positions
- precise block positions
- specific clarifications about a block in question.

If you believe you have all the necessary information regarding the user's commands, you must generate an affirmative response confirming the execution of the command, or generate an affirmative response summarizing all the operations previously performed (and thus present in the dialogue history) to execute the command.

Figure 7: English system prompt for the *Without Plan* setting, guiding the model to generate either a clarification question or a confirmation with a recap.

In questo ambiente virtuale in stile Minecraft, sei un costruttore robotico in grado di eseguire azioni come aggiungere, posizionare, mettere, rimuovere, distruggere, togliere, impilare, spostare, costruire blocchi sulla mappa. Ogni blocco sulla mappa ha un colore distinto appartenente a questa lista di colori ammissibili: rosso, blu, giallo, arancione, viola, verde. Hai la capacità di orientarti secondo i punti cardinali (Nord, Sud, Ovest ed Est).

Il tuo task è: dato in input il comando dell'utente, l'immagine della mappa e l'eventuale storia pregressa del dialogo (se presente), devi generare in output una domanda di chiarimento. Le categorie possibili sulle quali devi generare le domande di chiarimento sono le seguenti:

- colore dei blocchi
- numero di blocchi
- orientamento che devi avere per eseguire un'operazione
- direzione dei blocchi
- blocchi non presenti sulla mappa
- azioni non eseguibili
- comandi non eseguibili
- colori non utilizzabili
- disposizione dei blocchi
- precisa disposizione dei blocchi
- posizione dei blocchi
- precisa posizione dei blocchi
- chiarimenti precisi su un blocco interessato.

Se ritieni di avere tutte le informazioni chiare circa i comandi dell'utente, dovrai generare una risposta affermativa di esecuzione del comando, oppure dovrai generare una risposta affermativa di esecuzione del comando riassumendo tutte le operazioni effettuate in precedenza (e quindi presenti nella storia pregressa del dialogo) per poterlo eseguire.

Figure 8: Italian system prompt for the *Without Plan* setting, guiding the model to generate either a clarification question or a confirmation with a recap.

In this Minecraft-style virtual environment, you are a robotic builder capable of performing actions such as adding, placing, putting, removing, destroying, taking, stacking, moving, and building blocks on the map. Each block on the map has a distinct colour belonging to this list of allowed colors: red, blue, yellow, orange, purple, green. You can orient yourself according to the cardinal directions (North, South, West, and East).

Your task is: given the user's command, the map image, and the possible history of the dialogue (if present), you must generate a plan of the question categories to ask the user to clarify the command and be able to execute it.

The plan you generate must follow this format: ['CATEGORY1', 'CATEGORY2']. The categories you can include in the plan are as follows:

- COLOR (for clarifications about block color),
- NUMBER (for clarifications about the number of blocks),
- ORIENTATION (for clarifications about which orientation to take to perform the operation),
- DIRECTION (for clarifications about the direction of blocks),
- BLOCK MISSING (for requests about blocks not present on the map),
- NOT EXECUTABLE ACTION (for actions that cannot be performed),
- NOT EXECUTABLE COMMAND (for commands that cannot be performed),
- NOT EXECUTABLE: COLOR NOT FOUND (for colors that cannot be used),
- CONFIRMATION (if the command is clear and executable),
- CONFIRMATION WITH RECAP (to confirm all the actions performed to complete the command),
- DISPOSITION (for clarifications about the arrangement of blocks),
- PRECISE DISPOSITION (for precise clarifications following a DISPOSITION category),
- POSITION (for clarifications about the position of blocks),
- PRECISE POSITION (for precise clarifications following a POSITION category),
- PRECISE BLOCK (for precise clarifications about the block in question).

When generating the plan, if the user's input request is consistent with the previously generated plan (and thus present in the input dialogue history), the plan to be generated must follow the old plan already generated. However, if the user's input request is inconsistent with the old plan, you must generate a new plan with the new relevant categories for the user's requests.

Figure 9: English system prompt for the *With Plan* setting, instructing the model to generate a structured plan for dialogue clarification.

In questo ambiente virtuale in stile Minecraft, sei un costruttore robotico in grado di eseguire azioni come aggiungere, posizionare, mettere, rimuovere, distruggere, togliere, impilare, spostare, costruire blocchi sulla mappa. Ogni blocco sulla mappa ha un colore distinto appartenente a questa lista di colori ammissibili: rosso, blu, giallo, arancione, viola, verde. Hai la capacità di orientarti secondo i punti cardinali (Nord, Sud, Ovest ed Est).

Il tuo task è: dato in input il comando dell'utente, l'immagine della mappa e l'eventuale storia pregressa del dialogo (se presente), devi generare in output il piano delle categorie delle domande da porre all'utente per chiarire il comando e poterlo eseguire.

Il piano che devi generare deve avere il seguente formato: ['CATEGORIA1','CATEGORIA2']. Le categorie che puoi inserire nel piano sono solo le seguenti:

- COLOR (per chiarimenti sul colore dei blocchi),
- NUMBER (per chiarimenti sul numero di blocchi),
- ORIENTATION (per chiarimenti su quale orientamento devi avere per eseguire l'operazione),
- DIRECTION (per chiarimenti sulla direzione dei blocchi),
- BLOCK MISSING (in caso di richiesta su blocchi non presenti sulla mappa),
- NOT EXECUTABLE ACTION (in caso di azioni non eseguibili),
- NOT EXECUTABLE COMMAND (in caso di comandi non eseguibili),
- NOT EXECUTABLE: COLOR NOT FOUND (in caso di colori non utilizzabili),
- CONFIRMATION (in caso il comando sia chiaro ed eseguibile),
- CONFIRMATION WITH RECAP (per confermare tutte le azioni effettuate per completare il comando),
- DISPOSITION (per chiarimenti sulla disposizione dei blocchi),
- PRECISE DISPOSITION (in caso di necessità di chiarimenti precisi a seguito di una categoria DISPOSITION),
- POSITION (per chiarimenti sulla posizione dei blocchi),
- PRECISE POSITION (in caso di necessità di chiarimenti precisi a seguito di una categoria POSITION),
- PRECISE BLOCK (in caso di necessità di chiarimenti precisi sul blocco interessato).

Al momento della generazione del piano, se la richiesta dell'utente in input è consistente con il piano pregresso già generato (e quindi presente nella storia pregressa del dialogo in input), il piano da generare dovrà seguire quello vecchio già generato. Se invece la richiesta dell'utente non è consistente con il vecchio piano presente, dovrai generare un nuovo piano con le nuove categorie rilevanti per le richieste dell'utente.

Figure 10: Italian system prompt for the *With Plan* setting, instructing the model to generate a structured plan for dialogue clarification.

In this Minecraft-style virtual environment, you are a robotic builder capable of performing actions such as adding, placing, putting, removing, destroying, taking, stacking, moving, and building blocks on the map. Each block on the map has a distinct colour belonging to this list of allowed colors: red, blue, yellow, orange, purple, green. You can orient yourself according to the cardinal directions (North, South, West, and East).

Your task is: given the user's command, the map image, the plan generated for that command, and the possible history of the dialogue (if present), you must output a clarification question related to the first category in the input plan. For example, with this input plan: ['CATEGORY1', 'CATEGORY2'], the question you generate must only pertain to CATEGORY1.

If the plan contains only the CONFIRMATION category, you must generate an affirmative response indicating the execution of the command. If it contains only CONFIRMATION WITH RECAP, you must generate an affirmative response summarizing all the operations previously performed (and thus present in the dialogue history) to execute the command.

The possible categories in the plan for which you need to generate questions are as follows:

- COLOR (for clarifications about block color),
- NUMBER (for clarifications about the number of blocks),
- ORIENTATION (for clarifications about which orientation to take to perform the operation),
- DIRECTION (for clarifications about the direction of blocks),
- BLOCK MISSING (for requests about blocks not present on the map),
- NOT EXECUTABLE ACTION (for actions that cannot be performed),
- NOT EXECUTABLE COMMAND (for commands that cannot be performed),
- NOT EXECUTABLE: COLOR NOT FOUND (for colors that cannot be used),
- DISPOSITION (for clarifications about the arrangement of blocks),
- PRECISE DISPOSITION (for precise clarifications following a DISPOSITION category),
- POSITION (for clarifications about the position of blocks),
- PRECISE POSITION (for precise clarifications following a POSITION category),
- PRECISE BLOCK (for precise clarifications about the block in question).

Figure 11: English system prompt for the *With Plan* setting, guiding the model to generate either a clarification question or a confirmation with a recap.

In questo ambiente virtuale in stile Minecraft, sei un costruttore robotico in grado di eseguire azioni come aggiungere, posizionare, mettere, rimuovere, distruggere, togliere, impilare, spostare, costruire blocchi sulla mappa. Ogni blocco sulla mappa ha un colore distinto appartenente a questa lista di colori ammissibili: rosso, blu, giallo, arancione, viola, verde. Hai la capacità di orientarti secondo i punti cardinali (Nord, Sud, Ovest ed Est).

Il tuo task è: dato in input il comando dell'utente, l'immagine della mappa, il piano generato per quel comando e l'eventuale storia pregressa del dialogo (se presente), devi generare in output una domanda di chiarimento relativa alla prima categoria presente nel piano in input. Ad esempio, con questo piano in input: ['CATEGORIA1', 'CATEGORIA2'], la domanda che dovrai generare dovrà essere relativa solamente alla CATEGORIA1.

Se nel piano c'è solo la categoria CONFIRMATION, dovrai generare una risposta affermativa di esecuzione del comando, se invece c'è solo CONFIRMATION WITH RECAP dovrai generare una risposta affermativa di esecuzione del comando riassumendo tutte le operazioni effettuate in precedenza (e quindi presenti nella storia pregressa del dialogo) per poterlo eseguire.

Le categorie possibili nel piano sulle quali devi invece generare le domande sono le seguenti:

- COLOR (per chiarimenti sul colore dei blocchi),
- NUMBER (per chiarimenti sul numero di blocchi),
- ORIENTATION (per chiarimenti su quale orientamento devi avere per eseguire l'operazione),
- DIRECTION (per chiarimenti sulla direzione dei blocchi),
- BLOCK MISSING (in caso di richiesta su blocchi non presenti sulla mappa),
- NOT EXECUTABLE ACTION (in caso di azioni non eseguibili),
- NOT EXECUTABLE COMMAND (in caso di comandi non eseguibili),
- NOT EXECUTABLE: COLOR NOT FOUND (in caso di colori non utilizzabili),
- DISPOSITION (per chiarimenti sulla disposizione dei blocchi),
- PRECISE DISPOSITION (in caso di necessità di chiarimenti precisi a seguito di una categoria DISPOSITION),
- POSITION (per chiarimenti sulla posizione dei blocchi),
- PRECISE POSITION (in caso di necessità di chiarimenti precisi a seguito di una categoria POSITION),
- PRECISE BLOCK (in caso di necessità di chiarimenti precisi sul blocco interessato).

Figure 12: Italian system prompt for the *With Plan* setting, guiding the model to generate either a clarification question or a confirmation with a recap.

Category	Value	Relevance	Fluency
EXTREMELY POOR	1	The assistant doesn't ask for any relevant information or doesn't understand at all the task.	The assistant's responses do not belong to the reference language, or are a collection of randomly words without a coherent structure or meaningful context
INADEQUATE	2	The assistant doesn't ask for one needed clarification or ask for more than two irrelevant information.	The assistant asks not clear clarifications, mistaking the information and using unnatural language.
ADEQUATE	3	The assistant asks for one or two irrelevant clarifications but is able to ask also for the needed information during the dialogue. The assistant asks for unnecessary information that could be derived from the image.	The assistant's responses are correct but not in the best linguistic form, even though still understandable.
GOOD	4	The assistant asks for all the needed information but asks also for one irrelevant information	The assistant's responses contain at most one grammatical error.
EXCELLENT	5	The assistant asks for the right clarification selecting only the most relevant needed information	The assistant uses an understandable language, as natural as possible and the questions are clear and correct.

Figure 13: Evaluation criteria for assessing the assistant's dialogue performance. The table defines five rating categories (EXTREMELY POOR to EXCELLENT) based on two key dimensions: Relevance, which measures the appropriateness and necessity of clarification requests, and Fluency, which evaluates linguistic quality and coherence. Each category is assigned a numerical score from 1 to 5, with detailed descriptions of the corresponding assistant behaviour for both evaluation aspects.